Docket No.: 42P8534C

Confirmation No.: 6808

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**
**BOARD OF PATENT APPEALS AND INTERFERENCES**

| | |
|---|---|
| In re Application of:<br>Jin Yang<br>Application No. 10/666,619<br>Filed: September 17, 2003<br>For: SYMBOLIC MODEL CHECKING WITH DYNAMIC MODEL PRUNING | Examiner: Parihar, S.<br>Art Unit: 2825 |

**CERTIFICATE OF TRANSMISSION**

I hereby certify that this correspondence is being transmitted to the United States Patent and Trademark Office, or facsimile transmitted to Fax No. (531)273-8300

on 3-11-2008 /Lawrence M. Mennemeier/
Date Lawrence M. Mennemeier

**APPELLANT'S BRIEF UNDER 37 CFR § 41.37**
**IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES**

Mail Stop Appeal Brief-Patents
Commissioner of Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant hereby submits this Brief in support of an appeal from a non-final decision of the Examiner, in the above-referenced case. Appellant respectfully requests consideration of this appeal by the board of Patent Appeals and Interference for allowance of the above-referenced patent application.

TABLE OF CONTENTS

## I. Real Party in Interest

The real party in interest in the present appeal is Intel Corporation of Santa Clara, California, the assignee of the present application.

## II. Related Appeals and Interferences

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.

## III. Status of the Claims (independent claims shown in bold)

Claims **1**-5, **11**, and **16** stand rejected under 35 USC § 102 as allegedly being anticipated by M. Chiodo et al., "Automatic Compositional Minimization in CTL Model Checking," Nov. 1992, pp. 172-178 (Chiodo).

Claims 6-10 stand rejected under 35 USC § 112 as allegedly being indefinite.

Claims 6-10, 12-15 and 17-19 stand rejected under 35 USC § 103(a) as allegedly being unpatentable over Chiodo in view of US Patent Number 5,901,073 (Kurshan).

Non-final rejection of claims **1**-10, **11**-15 and **16**-19 is being appealed.

## IV. Status of Amendments

An official amendment and response to a first Office Action mailed 3/9/2006 was submitted by appellant on 9/11/2006 and was entered. A Final Office Action was mailed on 12/1/2006. Appellant responded by submitting an RCE with amendment and response on 6/1/2007, which was entered. A Non-final Office Action was mailed on 2/11/2008. A Notice of Appeal was transmitted on 2/11/2008, and an appeal ensued.

Accordingly, the claims stand as of Appellant's response of 6/1/2007, and are reproduced in clean form in the Claims Appendix.

## V. Summary of Claimed Subject Matter

Formal verification methods provide for improved efficiency of popular binary decision diagram (BDD) based algorithms. A forward assumption propagation method generates assumptions to characterize a set of interesting states for a property being evaluated at one or more evaluation stages. A dynamic transition relation reduction technique improves the efficiency for symbolic model checking by reducing transition relations under assumptions dynamically generated from properties being evaluated. In embodiments of the dynamic transition relation reduction technique, transition relations are pruned under assumptions dynamically generated from the logical structure of properties being evaluated. These methods provide symbolic model checking of circuits and other finite state systems previously too large to be completed successfully using BDD based algorithms.

Claim 1, for example, sets forth a computer software product for formal verification of circuits or other finite-state systems, the computer software product having one or more recordable medium having executable instructions stored thereon which, when

executed by a processing device, causes the processing device to: generate, from a first property, a first assumption[1] including a first state predicate; generate, for a model, a first transition relation[2] that includes the first state predicate; and reduce the first transition relation according to the first assumption[2,3].

---

[1] "Given a parsing of the property, assumptions may be generated for the sup-properties to be evaluated. Figure 5b illustrates one embodiment of a method for producing and propagating assumptions from sub-properties to be evaluated. In processing block 521, the assumption for iteration zero, $Assum_0$, is initialized to the value one (true) and Node is set to the root 500 of the property to be evaluated. The iteration counter i is then incremented and processing proceeds to processing block 522. In processing block 522, the Node is tested to see if it consists of a single variable, in which case processing terminates at processing block 522. If not, processing proceeds to processing block 523. In processing block 523 the type of the Node operation is identified. If it is an implication operation processing proceeds at processing block 524. On the other hand, if it is a next state operator X, then processing proceeds to processing block 525.
In processing block 524 the assumption for iteration i, $Assum_i$, is set to the assumption for iteration i-1, $Assum_{i-1}$, combined with the left sub-property of Node using the logical AND operation. In processing block 525 the assumption for iteration i, $Assum_i$, is set to post-image of the assumption for iteration i-1, $Assum_{i-1}$. Processing then proceeds to processing block 526 from processing block 524 or from processing block 525, where Node is set to the right sub-property of Node. The iteration counter i is then incremented and processing proceeds to processing block 522.
Figure 5c shows an example of producing and propagating assumptions from sub-properties to be evaluated. In iteration zero, assumption 540 is set to the value one and sub-property 530 is set to the state predicate for property to be evaluated $(a \Rightarrow (b \Rightarrow X(Xf)))$. In iteration one, assumption 541 is set to a.=.(1 AND a) in accordance with processing block 524 and sub-property 531 is set to the right sub-property of sub-property 530, $b \Rightarrow X(Xf)$ in accordance with processing block 526. In iteration two, assumption 542 is set to a AND b and sub-property 532 is set to the right sub-property of sub-property 531, X(Xf). In iteration three, assumption 543 is set to Post(a AND b), which may be evaluated as d since N(d) = b,.and sub-property 533 is set to the right sub-property of sub-property 532, Xf. In iteration four, assumption 544 is set to Post(Post(a.AND.b).which may be evaluated to one (true) and sub-property 534 is set to the right sub-property of sub-property 533, f.
The number of variables in a transition relation may be reduced according to a dynamically generated assumption as the transition relation is built. For instance the next state function for a variable f, N(f) = c NAND d may be pruned according to an assumption including (d=1) to N(f) = NOT c." **(p. 17, line 7 through p. 18, line 21, Figs. 5b & 5c)**

[2] "The number of variables in a transition relation may be reduced according to a dynamically generated assumption as the transition relation is built. For instance the next state function for a variable f, N(f)=c NAND d may be pruned according to an assumption including (d=1) to N(f)=NOT c.
FIG. 6a graphically illustrates the state space of a model with a transition relation that was built according to one embodiment of an iterative lazy pre-image computation method to evaluate sub-property 530 or 534. It includes five variables that may be exhaustively searched for an assignment that satisfies the sub-property 530 or 534. These variables are f at block 611, c at block 613, d at block 615, a at block 617 and b at block 618. The three internal variables for which next state functions are included in the transition relation are, N(f) at block 612, N(c) at block 614, and N(d) at block 616. Since the assumptions 540 and 544 are trivial no pruning may be performed on the next state functions.
FIG. 6b graphically illustrates the state space of a dynamically pruned model with a transition relation that was built according to one embodiment of a lazy pre-image computation method to evaluate sub-property 531 under assumption 541. It includes four variables that may be exhaustively searched for an assignment that satisfies the sub-properties 531 or 532. These variables are f at block 621, c at block 623, d at block 625, and b at block 628. The three internal variables for which next state functions are included in the transition relation are, N(f) at block 622, N(c) at block 624, and N(d) at block 626. Since the assumption

Claim 6 sets forth the computer software product recited in Claim 1 wherein the first assumption is generated from an implication structure of the first property [1,4].

---

541 includes only the input variable, a, pruning of the transition relation may be performed only on the next state function for c, producing N(c)=c instead of N(c)=c AND a.
FIG. 6c graphically illustrates the state space of a dynamically pruned model with a transition relation that was built according to one embodiment of a lazy pre-image computation method to evaluate sub-property 532 under assumption 542. It includes three variables that may be exhaustively searched for an assignment that satisfies the sub-property 531. These variables are f at block 621, c at block 623, d at block 625. The three internal variables for which next state functions are included in the transition relation are, N(f) at block 622, N(c) at block 624, and N(d) at block 626. Since the assumption 541 includes the input variables, a and b, pruning of the transition relation may be performed on the next state function for c, producing N(c)=c instead of N(c)=c AND a and on the next state function for d, producing N(d)=1 instead of N(d)=b.
FIG. 6d graphically illustrates the state space of a dynamically pruned model with a transition relation that was built according to one embodiment of a lazy pre-image computation method to evaluate sub-property 533 under assumption 543. It includes only two variables that may be exhaustively searched for an assignment that satisfies the sub-property 533. These variables are f at block 631 and c at block 633. Both of these internal variables' next state functions are included in the transition relation. They are, N(f) at block 632, and N(c) at block 634. Since the assumption 543 includes the input variable, a, and the internal variable, d, pruning of the transition relation may be performed on the next state functions for c, producing N(c)=c instead of N(c)=c AND a and for f, producing N(f)=NOT c instead of N(f)=c NAND d." **(p. 18, line 18 through p. 20, line 17; Figs. 6a-6d)**

[3] "FIG. 7b shows an example of dynamically pruning a transition relation as it is built in a lazy pre-image computation according to assumptions generated from sub-properties to be evaluated. In iteration 4, state predicate 724 is set to the singular Node variable, f, and evaluated under the assumption 1 (true) which leaves the predicate unchanged. In iteration 3, state predicate 723 is set to the lazy pre-image of the predicate of the predicate f and evaluated under the assumption, d, which in accordance with processing block 716, reduces the next state function N(f)=c NAND d to N(f)=NOT c. In iteration 2, state predicate 722 is set to the lazy pre-image of the predicate, NOT c, and evaluated under the assumption, a AND b, which in accordance with processing block 716, reduces the next state function N(c)=a AND c to N(c)=c. In iteration 1, state predicate 721 is set to the logical OR combination of predicate, NOT c, and negated left sub-property, b, resulting in (NOT b OR NOT c) and then evaluated under the assumption, a, in accordance with processing block 715, which leaves the predicate unchanged. In iteration 0, state predicate 720 is set to the logical OR combination of predicate, (NOT b OR NOT c), and negated left sub-property, a, resulting in (NOT a OR NOT b OR NOT c) and then evaluated under the assumption, a, in accordance with processing block 715, which leaves the predicate unchanged." **(p. 23, lines 3-21; Fig. 7b)**

[4] "Figure 5a illustrates, for example, a parsing of a property 510, $a \Rightarrow (b \Rightarrow X(Xf))$ in accordance with one corresponding logical implication structure of property 510. At the first stage the property is parsed into a root 500 representing the logical implication operation, a left sub-property 506 representing the variable, a, and a right sub-property 511 representing $b \Rightarrow X(Xf)$." **(p. 16, lines 19-23, Fig. 5a)**
"At the second stage the sub-property 511 is parsed into a root 501 representing the logical implication operation, a left sub-property 507 representing the variable, b, and a right sub-property 512 representing X(Xf). At the third stage the property is parsed into a root 502 representing the next state operator X, and a right sub-property 513 representing Xf. Finally at the fourth stage the sub-property 513 is parsed into a root 503 representing the next state operator X, and a right sub-property 514 representing f. Given a parsing of the property assumptions may be generated for the sup-properties to be evaluated." **(p. 17, lines 1-8, Fig. 5a)**
"In processing block 524 the assumption for iteration i, $Assum_i$, is set to the assumption for iteration i-1, $Assum_{i-1}$, combined with the left sub-property of Node using the logical AND operation. In processing block 525 the assumption for iteration i, $Assum_i$, is set to post-image of the assumption for iteration i-1, $Assum_{i-1}$. Processing then proceeds to processing block 526 from processing block 524 or from processing block 525, where Node is set to the right sub-property of Node. The iteration counter i is then incremented and processing proceeds to processing block 522." **(p. 17, line 20 through p. 18, line 4; Fig. 5b)**

Similarly Claim 17 sets forth the verification system of Claim 16 wherein the first assumption is produced from the logical structure of the first property[1,4].

VI. Grounds of Rejection to be Reviewed on Appeal

A. Claims 6-10 stand rejected under 35 USC § 112 as allegedly being indefinite.

B. Claims 1-5, 11, and 16 stand rejected under 35 USC § 102 as allegedly being anticipated by M. Chiodo et al., "Automatic Compositional Minimization in CTL Model Checking," Nov. 1992, pp. 172-178 (Chiodo).

C. Claims 6-10, 12-15 and 17-19 stand rejected under 35 USC § 103(a) as allegedly being unpatentable over Chiodo in view of US Patent No. 5,901,073 (Kurshan).

VII. Argument

A. 35 U.S.C. § 112 REJECTIONS

Claims 6-10 stands rejected under 35 USC § 112, second paragraph, as allegedly being indefinite for failing to point out and distinctly claim the invention.

Claims 6-10 Are Not Indefinite.

With regard to Claim 6 the Office Action of Aug. 9, 2007 states (pp. 2-3, No. 4) that the phrase "implication structure" is not is not clearly described in the specification, that it is not clearly defined in Claim 6, and that what is implied or suggested by the "implication structure" is not described. Appellant respectfully disagrees.

The issue of definiteness is whether, in light of the teachings of the prior art and of the particular invention, the claims set out and circumscribe a particular area with a reasonable degree of precision and particularity. *In re Moore*, 439 F.2d 1232, 1235, 169 USPQ 236, 238 (CCPA 1971).

Claim 6, for example, sets forth (emphasis added):

> 6. (Original) The computer software product recited in Claim 1 wherein the first assumption is generated from an implication structure of the first property.

With regard to what is implied or suggested by the "implication structure," Appellant respectfully submits that one skilled in the art would understand the implication operation of Boolean logic as taught by the prior art, and would understand that such operations and their arguments may be representable as data structures, e.g. such as a tree having a root, a left subtree and a right subtree. Any Boolean logic formula may be translated into an equivalent formula that uses only implication and negation, a fact, of which is known to those of skill in the art.

The test for definiteness under 35 U.S.C. § 112 is whether those skilled in the art would understand what is claimed when the claim is read in light of the specification. *Orthokinetics, Inc. v. Safety Travel Chairs, Inc.*, 806 F.2d 1565, 1576, 1 USPQ2d, 1081, 1088 (Fed. Cir. 1986).

Appellant respectfully submits that the specification has set forth a full and clear description of the claimed subject matter. For example, with regard to an implication structure of the first property, the specification discloses (e.g. on p. 16, lines 19-23, Fig. 5a; emphasis added) that:

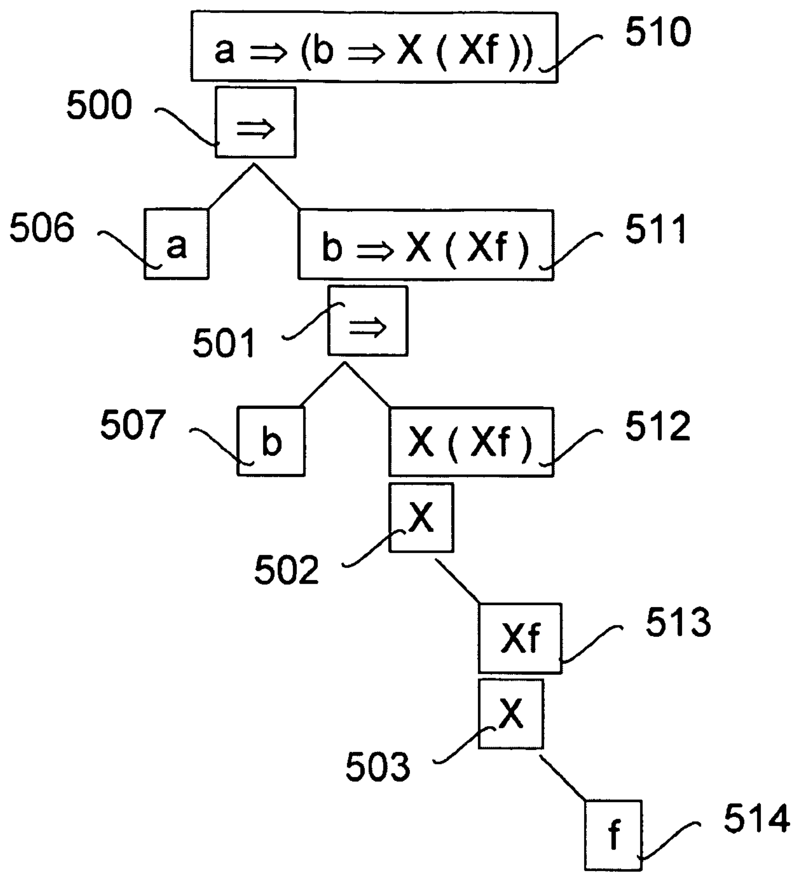
> Figure 5a illustrates, for example, a parsing of a property 510, a ⇒ (b ⇒ X(Xf)) in accordance with one corresponding logical implication structure of property 510. At the first stage the property is parsed into a root 500 representing the logical implication operation, a left sub-property 506 representing the variable, a, and a right sub-property 511 representing b ⇒ X(Xf).

and further discloses (e.g. on p. 17, lines 1-8, Fig. 5a; emphasis added) that:

> At the second stage the sub-property 511 is parsed into a root 501 representing the logical implication operation, a left sub-property 507 representing the variable, b, and a right sub-property 512 representing X(Xf). At the third stage the property is parsed into a root 502 representing the next state operator X, and a right sub-property 513 representing Xf. Finally at the fourth stage the sub-property 513 is parsed into a root 503 representing the next state operator X, and a right sub-property 514 representing f. Given a parsing of the property assumptions may be

generated for the sup-properties to be evaluated.

The present specification also illustrates an example structure in Fig. 5a as:




The present specification further discloses (p. 17, line 20 through p. 18, line 4; Fig. 5b; emphasis added) that:

> In processing block 524 the assumption for iteration i, Assum$_i$, is set to the assumption for iteration i-1, Assum$_{i-1}$, combined with the left sub-property of Node using the logical AND operation. In processing block 525 the assumption for iteration i, Assum$_i$, is set to post-image of the assumption for iteration i-1, Assum$_{i-1}$. Processing then proceeds to processing block 526 from processing block 524 or from processing block 525, where Node is set to the right sub-property of Node. The iteration counter i is then incremented and processing proceeds to processing block 522.

Appellant respectfully submits that a person of skill in the field, having read and understood the specification would understand what is claimed by generating the first assumption "from an implication structure of the first property" as set forth in Claim 6.

The Federal Circuit explained in *Multiform Desiccants, Inc. v. Medzam, Ltd., 133 F.3d 1473, 1477 (Fed. Cir. 1998)*:

> It is the person of ordinary skill in the field of the invention through whose eyes the claims are construed. Such person is deemed to read the words used in the patent documents with an understanding of their meaning in the field, and to have knowledge of any special meaning and usage in the field. The inventor's words that are used to describe the invention--the inventor's lexicography--must be understood and interpreted by the court as they would be understood and interpreted by a person in that field [**25] of technology. Thus the court starts the decision making process by reviewing the same resources as would that person, viz., the patent specification and the prosecution history.

Appellant respectfully submits that at least in light of the above disclosure set forth by the specification, Claim 6 sets out and circumscribes generating the first assumption from an implication structure of the first property, with a reasonable degree of precision and particularity required by 35 USC § 112, second paragraph.

B. <u>35 U.S.C. § 102 REJECTIONS</u>

Claims **1-5**, **11**, and **16** stand rejected under 35 USC § 102 as allegedly being anticipated by M. Chiodo et al., "Automatic Compositional Minimization in CTL Model Checking," Nov. 1992, pp. 172-178 (Chiodo).

<u>Claims **1-5**, **11**, and **16** Are Not Anticipated.</u>

Appellant submits that in order for a rejection based on anticipation to be maintained, the identical invention must be shown in as complete detail as is contained in the claim. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Claim 1, for example, sets forth:

> 1. (Previously Presented) A computer software product for formal verification of circuits or other finite-state systems, the computer software product having one or more recordable medium having executable instructions stored thereon which, when executed by a processing device, causes the processing device to:
> generate, from a first property, a first assumption including a first state predicate;
> generate, for a model, a first transition relation that includes the first state predicate; and
> reduce the first transition relation according to the first assumption.

and Claim 11 sets forth:

> 11. (Previously Presented) A verification system for verification of circuits or other finite-state systems, the verification system comprising:
> means for producing, from a first property, a first assumption including a first state predicate; and
> means for producing a reduced next state function from a first next state function involving the first state predicate by applying the first assumption.

With regard to Claims 1, 11 and 16, the Office Action of Aug. 9, 2007 asserts (pp. 3 and 4-5, Nos. 6, 11 and 12) that Chiodo teaches generating or producing an assumption from a first property; and reducing a transition relation or producing a

reduced next state function according to that assumption. Appellant respectfully disagrees.

Chiodo describes two categories of methods to avoid the construction of a complete state graph for a finite state system, which he refers to as *compositional verification* and *compositional minimization*. In *compositional verification*, Chiodo describes that "one tries to deduce properties of a composition of processes by reasoning on the individual components and their interactions without ever building the composed system." (Chiodo, p. 172, Introduction, par. 2).

Chiodo is directed to a *compositional minimization* method he calls *compositional model checking*, which uses a *restrict operator* as a heuristic and hopes that the BDD size of the product of reduced machines will be smaller than the complete product machine. Chiodo notes that, "Only through experimentation will indicate the effectiveness of this heuristic." (p. 175, §§3.2.2, lines 12-22)

The Examiner claims (p. 3, No. 6; p. 4, No. 11; p. 5, No. 12) that generating or producing an assumption from a first property is anticipated by Chiodo where, one deduces properties by reasoning in the *compositional verification* method.

But Appellant respectfully submits that the examples given by Chiodo for this category, are: Wolper, who inductively verifies systems by looking for invariants; Kurshan and McMillan, who attempted a similar approach; and Grumberg, who defines subsets of the logics where only universal path quantification is allowed. (p. 172, Introduction, par. 3) These examples are apparently deductions of properties by human reasoning on components without ever building the composed system.

Appellant respectfully submits that the claim language must be construed through

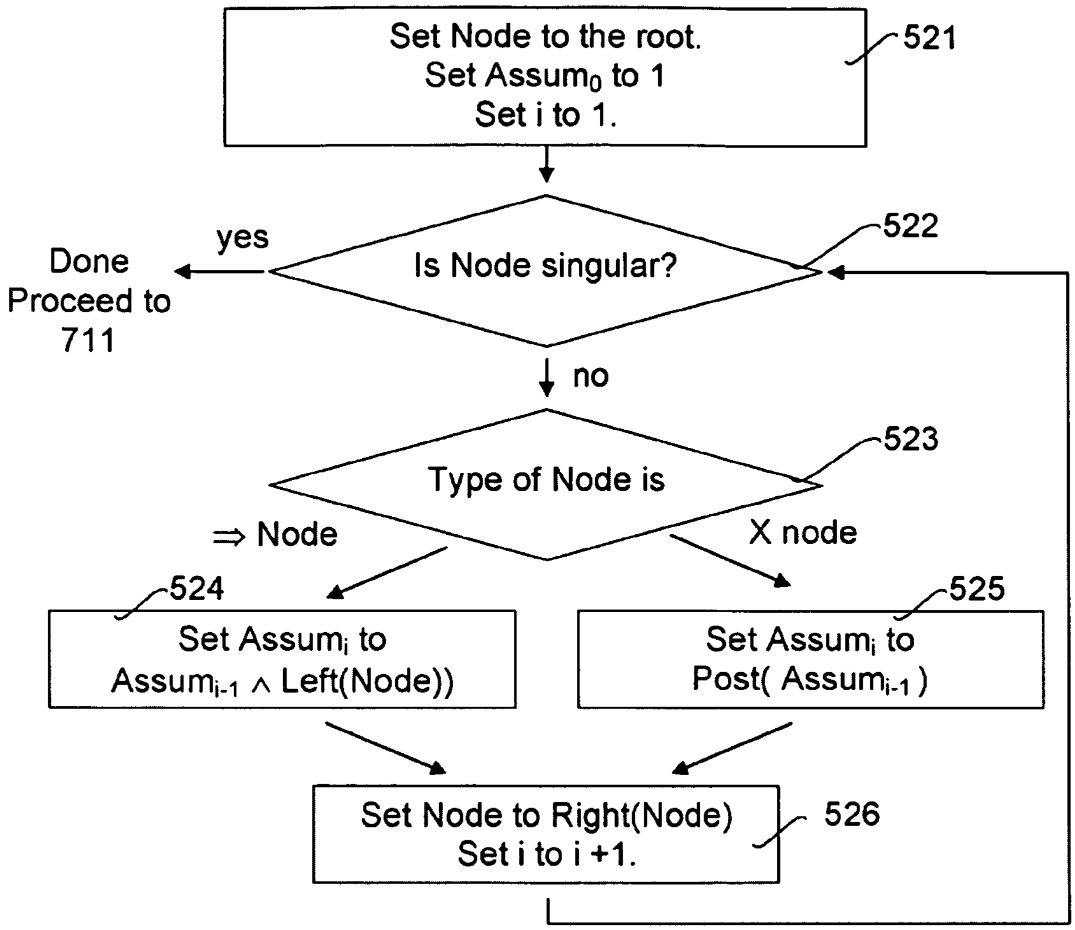
the eyes of a person of ordinary skill in the art who is deemed to read the claim in the context of the entire patent, including the specification.

It was explained recently in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313; 75 U.S.P.Q.2D (BNA) 1321 (Fed. Cir. 2005): Importantly, the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification.

The present specification has disclosed the generation by a processing device, from a first property, of a first assumption including a first state predicate, for example (Figs. 5b & 5c; p. 17, line 7 through p. 18, line 4) as:

> Given a parsing of the property, assumptions may be generated for the sup-properties to be evaluated.
>
> Figure 5b illustrates one embodiment of a method for producing and propagating assumptions from sub-properties to be evaluated. In processing block 521, the assumption for iteration zero, $Assum_0$, is initialized to the value one (true) and Node is set to the root 500 of the property to be evaluated. The iteration counter i is then incremented and processing proceeds to processing block 522. In processing block 522, the Node is tested to see if it consists of a single variable, in which case processing terminates at processing block 522. If not, processing proceeds to processing block 523. In processing block 523 the type of the Node operation is identified. If it is an implication operation processing proceeds at processing block 524. On the other hand, if it is a next state operator X, then processing proceeds to processing block 525.
>
> In processing block 524 the assumption for iteration i, $Assum_i$, is set to the assumption for iteration i-1, $Assum_{i-1}$, combined with the left sub-property of Node using the logical AND operation. In processing block 525 the assumption for iteration i, $Assum_i$, is set to post-image of the assumption for iteration i-1, $Assum_{i-1}$. Processing then proceeds to processing block 526 from processing block 524 or from processing block 525, where Node is set to the right sub-property of Node. The iteration counter i is then incremented and processing proceeds to processing block 522.

FIG. 5b




The present specification further discloses (p. 18, lines 5-21, Fig. 5c) that:

Figure 5c shows an example of producing and propagating assumptions from sub-properties to be evaluated. In iteration zero, assumption 540 is set to the value one and sub-property 530 is set to the state predicate for property to be evaluated (a ⇒ (b ⇒ X(Xf)) ). In iteration one, assumption 541 is set to a.=.(1 AND a) in accordance with processing block 524 and sub-property 531 is set to the right sub-property of sub-property 530, b ⇒ X(Xf) in accordance with processing block 526. In iteration two, assumption 542 is set to a AND b and sub-property 532 is set to the right sub-property of sub-property 531, X(Xf). In iteration three, assumption 543 is set to Post(a AND b), which may be evaluated as d since N(d) = b,.and sub-property 533 is set to the right sub-property of sub-property 532, Xf. In iteration four, assumption 544 is set to Post(Post(a.AND.b).which may be evaluated to one (true) and sub-property 534 is set to the right sub-property of sub-property 533, f.

The number of variables in a transition relation may be reduced according to a dynamically generated assumption as the transition relation is built. For instance the next state function for a variable f, N(f) = c NAND d may be pruned according to an assumption including (d=1) to N(f) = NOT c.

Appellant respectfully submits that causing a processing device to generate, from

a first property, a first assumption including a first state predicate, as set forth in Claim 1, would not be construed in the context of the specification, by one skilled in the art, to include the apparent deduction of properties by human reasoning on components without ever building the composed system, as presented by Chiodo.

Appellant further submits that Chiodo does not disclose or suggest using these apparent deductions of properties by human reasoning to reduce a transition relation or to produce a reduced next state function in his *compositional minimization* method.

Therefore, Appellant respectfully submits that, Chiodo does not disclose generating or producing a first assumption from a first property by a processing device and reducing a transition relation or producing a reduced next state function according to the first assumption in as complete detail as is set forth in Claims 1, 11, and 16.

C. <u>35 U.S.C. § 103 REJECTIONS</u>

Claims 6-10, 12-15 and 17-19 stand rejected under 35 USC § 103(a) as allegedly being unpatentable over Chiodo in view of US Patent Number 5,901,073 (Kurshan).

<u>Claims 6-10, 12-15 and 17-19 Are Not Obvious.</u>

With regard to Claims 6, 12 and 17, the Office Action of Aug. 9, 2007 claims (p. 6, No. 15) that Kurshan teaches an assumption being generated from an implication structure, or from a logical structure of a first property; and that it would have been obvious to incorporate Kurshan into the invention of Chiodo to provide a method for reducing the computational resources required to formally verify a system design. Appellant respectfully disagrees.

First, in determining the scope and content of the cited references with regard to the instant claims at issue, Appellant respectfully submits that Chiodo describes two categories of methods to avoid the construction of a complete state graph for a finite state system, which he refers to as *compositional verification* and *compositional minimization*. In *compositional verification*, Chiodo describes that "one tries to deduce properties of a composition of processes by reasoning on the individual components and their interactions without ever building the composed system." (Chiodo, p. 172, Introduction, par. 2).

Chiodo is directed to a *compositional minimization* method he calls *compositional model checking*, which uses a *restrict operator* as a heuristic and hopes that the BDD size of the product of reduced machines will be smaller than the complete product machine.

Chiodo notes that, "Only through experimentation will indicate the effectiveness of this heuristic." (p. 175, §§3.2.2, lines 12-22)

The Examiner claims (p. 3, No. 6; p. 4, No. 11; p. 5, No. 12) that generating or producing an assumption from a first property is anticipated by Chiodo where, one deduces properties by reasoning in the *compositional verification* method. Thus in the context of Chiodo, the "assumption" presumably means a condition or assertion that is believed to be true through reasoning without building the composed system to demonstrate it.

As the Office Action of Aug. 9, 2007 admits, Chiodo does not disclose or suggest an assumption being generated from an implication structure, or from a logical structure of a first property as set forth in Claims 6, 12 and 17. (p. 6, No. 15, par. 2)

Kurshan is directed to restricting the range of assumable values of variables and inputs of a system model to a restricted set of values, based on a partial search or partial verification of the system model. (Abstract, lines 6-10) This is accomplished by restricting the range of assumable values of a set of system model variables and system model inputs, based on the behavior of the system model during a partial search. A partial search is a search which identifies the states which the system model can enter when given a partial set of input values. That is, a partial search does not test the behavior of the system model in response to every possible input value (i.e. the entire range of assumable values). Rather, during a partial search, the system model inputs assume only a portion of the total range of assumable values, not every value in the assumable range of values. In general the values assumed by each system model variable or system model input during the partial search are used to define a restricted set of assumable values for

that variable or input, wherein each restricted set of assumable values defines the values that the particular system model input or variable can assume during system model verification. (Summary of the Invention, col. 3, lines 14-32)

Next, Appellant respectfully points out some of the differences between the cited references and the instant claims at issue. Claim 6, for example, sets forth:

> 6. (Original) The computer software product recited in Claim 1 wherein the first assumption is generated from an implication structure of the first property.

And Claim 17 sets forth:

> 17. (Original) The verification system of Claim 16 wherein the first assumption is produced from the logical structure of the first property.

The Office Action of Aug. 9, 2007 claims (p. 6, No. 15, par. 3) Kurshan teaches that an assumption is being generated from an implication structure, or from a logical structure of a first property because the assumed values are generated as a result of assumable values, citing Figure 2 of Kurshan, and that assumable values are implied values.

But, Appellant respectfully submits that *assume*, in the context of Kurshan, means to take on a value (i.e. synonymous with *assign*) so the assumable values are the values that can be assigned to variables or inputs of a system model. Thus the values assumed by each system model variable or system model input during the partial search are used to define a restricted set of assumable values for that variable or input. (col. 3, lines 26-29)

Therefore, Kurshan does not disclose or suggest that an assumption is being generated from an implication structure, but rather only that the values that can be assigned to variables is being restricted to the values that were assigned to them during a partial search.

Appellant respectfully submits that the Examiner is in error for misconstruing the instant terms outside of their respective contexts to give them new meanings, and in such a way as they would not be construed within their respective contexts by one of ordinary skill in the art. The terms "assumed" and "assumable" as used by Kushan have similarities in spelling with the term "assumption" as set forth by Claims 6, 12 and 17, but insufficiently similar meanings when properly construed in context.

Moreover, the restricted set of assigned values during a partial search of Kushan can not be construed as the same thing as a condition or assertion that is taken to be true through apparent human reasoning as described by Chiodo. Nor would the assumable values of Kushan be construed as the implication structure or logical structure of the first property, by one of skill in the art, in the context of the entire patent, including the specification.

Accordingly in light of the above arguments, Claims 6-10, 12-15 and 17-19 are not obvious in view of the cited references.

Conclusion

Appellant submits that all claims now pending are in condition for allowance. Such action is earnestly solicited at the earliest possible date. If there is a deficiency in fees, please charge our Deposit Acct. No. 50-0221.

Respectfully submitted,

Date: March 11, 2008

/Lawrence M. Mennemeier/
Lawrence M. Mennemeier
Reg. No. 51,003

INTEL CORPORATION
c/o INTELLEVATE LLP
P.O. Box 52050
Minneapolis, MN 55402
(408) 765-2194

VIII. Claims Appendix: Claims Allowed and Involved in Appeal (Clean Copy)

1. (Previously Presented) A computer software product for formal verification of circuits or other finite-state systems, the computer software product having one or more recordable medium having executable instructions stored thereon which, when executed by a processing device, causes the processing device to:

   generate, from a first property, a first assumption including a first state predicate;

   generate, for a model, a first transition relation that includes the first state predicate; and

   reduce the first transition relation according to the first assumption.

2. (Original) The computer software product recited in Claim 1 wherein reducing the first transition relation reduces the size of the model.

3. (Original) The computer software product recited in Claim 1 wherein reducing the first transition relation reduces the computational complexity of evaluating the first property.

4. (Original) The computer software product recited in Claim 1 wherein reducing the first transition relation reduces the number of variables in the model.

5. (Original) The computer software product recited in Claim 1 wherein reducing the first transition relation reduces the number of variables in the first transition relation.

6. (Original) The computer software product recited in Claim 1 wherein the first assumption is generated from an implication structure of the first property.

7. (Original) The computer software product recited in Claim 6 which, when executed by a processing device, further causes the processing device to:

propagate the first assumption to generate a second assumption according to a second property.

8. (Original) The computer software product recited in Claim 7 wherein the second property is a sub-property of the first property.

9. (Original) The computer software product recited in Claim 7 wherein the second property is to be evaluated under the first assumption.

10. (Original) The computer software product recited in Claim 7 wherein the first assumption is propagated only one transition stage to generate the second assumption.

11. (Previously Presented) A verification system for verification of circuits or other finite-state systems, the verification system comprising:

means for producing, from a first property, a first assumption including a first state predicate; and

means for producing a reduced next state function from a first next state function involving the first state predicate by applying the first assumption.

12. (Original) The verification system of Claim 11 wherein the first assumption is produced from the structure of the first property.

13. (Original) The verification system of Claim 12 further comprising:

means for propagating the first assumption according to a second property to generate a second assumption; and

means for producing, for a model, a transition relation that includes the reduced next state function.

14. (Original) The verification system of Claim 13 wherein the second property is a sub-property of the first property.

15. (Original) The verification system of Claim 14 wherein the first assumption is propagated only one transition stage to generate the second assumption.

16. (Previously Presented) A verification system for verification of circuits or other finite-state systems, the verification system comprising:

a recordable medium to store executable instructions;

a processing device to execute executable instruction; and

a plurality of executable instructions to cause the processing device to:

produce, from a first property, a first assumption including a first state predicate;

produce, for a model, a first transition relation that includes the first state predicate; and

reduce the first transition relation according to the first assumption.

17. (Original) The verification system of Claim 16 wherein the first assumption is produced from the logical structure of the first property.

18. (Original) The verification system recited in Claim 17, the plurality of executable instructions further comprising instructions to cause the processing device to:

propagate the first assumption to generate a second assumption according to a second state predicate.

19. (Previously Presented) The verification system recited in Claim 18 wherein the second property is a sub-property of the first property.

## IX. Evidence Appendix: With Copies of Evidence Relied Upon by Appellant

Appellant relies upon no additional evidence in this appeal.

## X. Related Proceedings Appendix: Copies of Decisions Rendered by a Court or the Board in any Prior and Pending Appeals, Interferences or Judicial Proceedings

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.